

# (12) UK Patent Application (19) GB (11) 2 276 255 (13) A

(43) Date of A Publication 21.09.1994

(21) Application No 9403001.2

(22) Date of Filing 17.02.1994

(30) Priority Data

(31) 08018680 (32) 17.02.1993 (33) US

(71) Applicant(s)

Andor International Ltd

(Incorporated in Bermuda)

Gibbons Building, 10 Queen Street, Hamilton HM EX,  
Bermuda

(72) Inventor(s)

James M. Bodwin

David J King

William G. Nadzam

Charles H. Reinking

(51) INT CL<sup>5</sup>

G06F 11/00

(52) UK CL (Edition M)

G4A AEX AME

(56) Documents Cited

GB 2202976 A

(58) Field of Search

UK CL (Edition M) G4A AEX AME

INT CL<sup>5</sup> G06F 11/00 11/08 11/10 11/14

ONLINE DATABASES : WPI

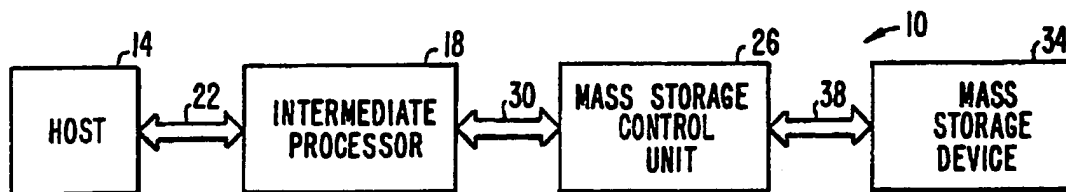
(74) Agent and/or Address for Service

R G C Jenkins &amp; Co

26 Caxton Street, LONDON, SW1H 0RJ,  
United Kingdom**(54) Track buffer with embedded error data**

(57) A buffer stores a plurality of records read from a mass storage device. The buffer includes a record information field and an error flag field for each stored record. The record information field includes count, key and data fields, together with an optional error data field for storing error data indicating a characteristic of an error which occurred when the record was read from the mass storage device. The error data field is typically located adjacent to the field which caused the error. The error flag field stores information indicating whether an error occurred when the record was read from the mass storage device. The error flag field also indicates whether the error was a permanent error. When the host processor requests from the buffer a record which has an error associated with it, the host is informed that the error occurred and is provided with the error data and the record data. If the error was a permanent error, then a unit check is provided to the host for indicating that the record data is not valid.

The invention allows the host processor to be aware that the data in the cache buffer contains corrected errors, and also means that host processor need not waste time trying to retrieve data that the cache has already failed to retrieve.

**FIG. 1.****GB 2 276 255 A**

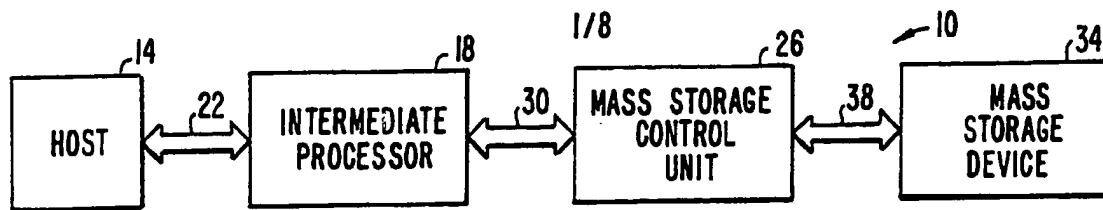


FIG. 1.

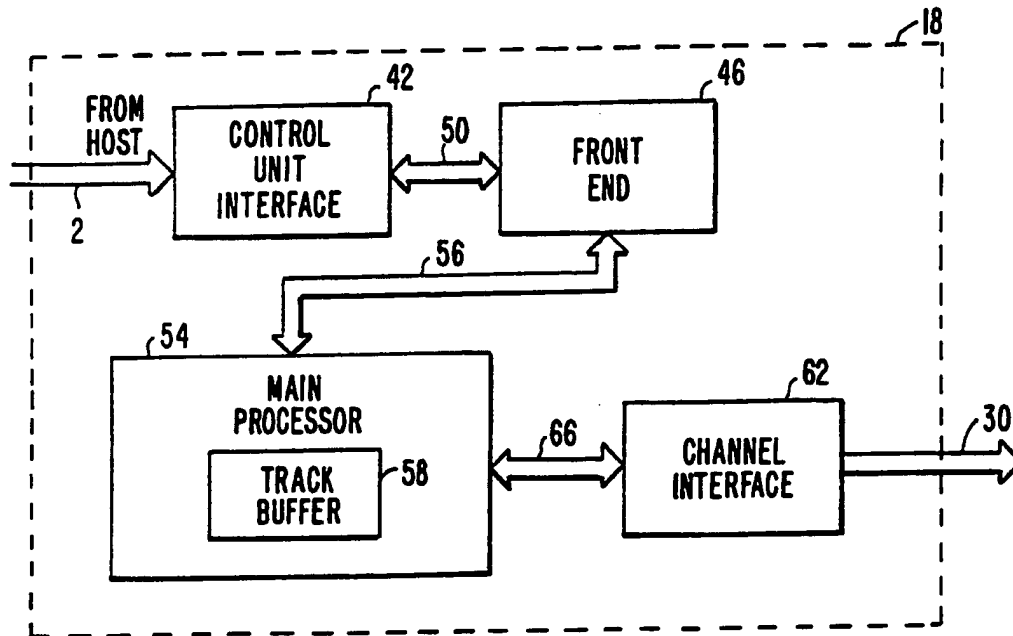


FIG. 2.

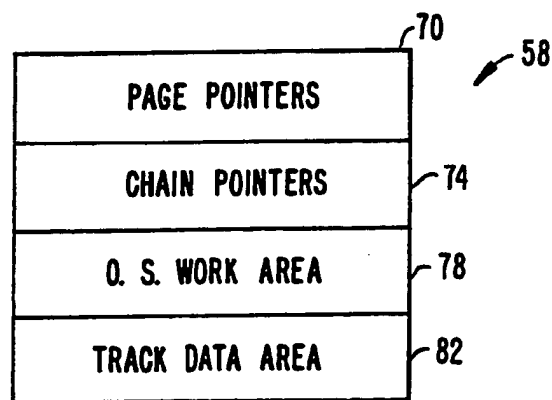


FIG. 3.

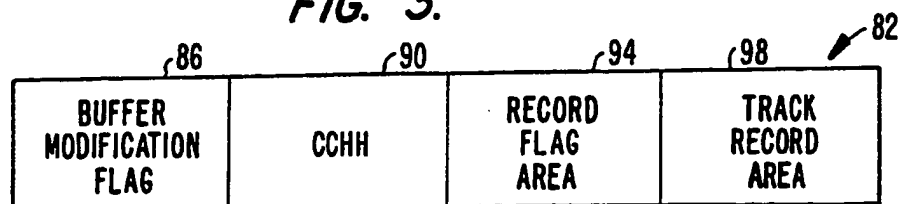


FIG. 4.

2/8

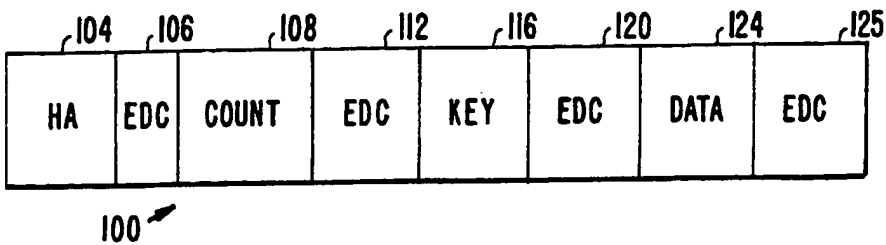


FIG. 5.

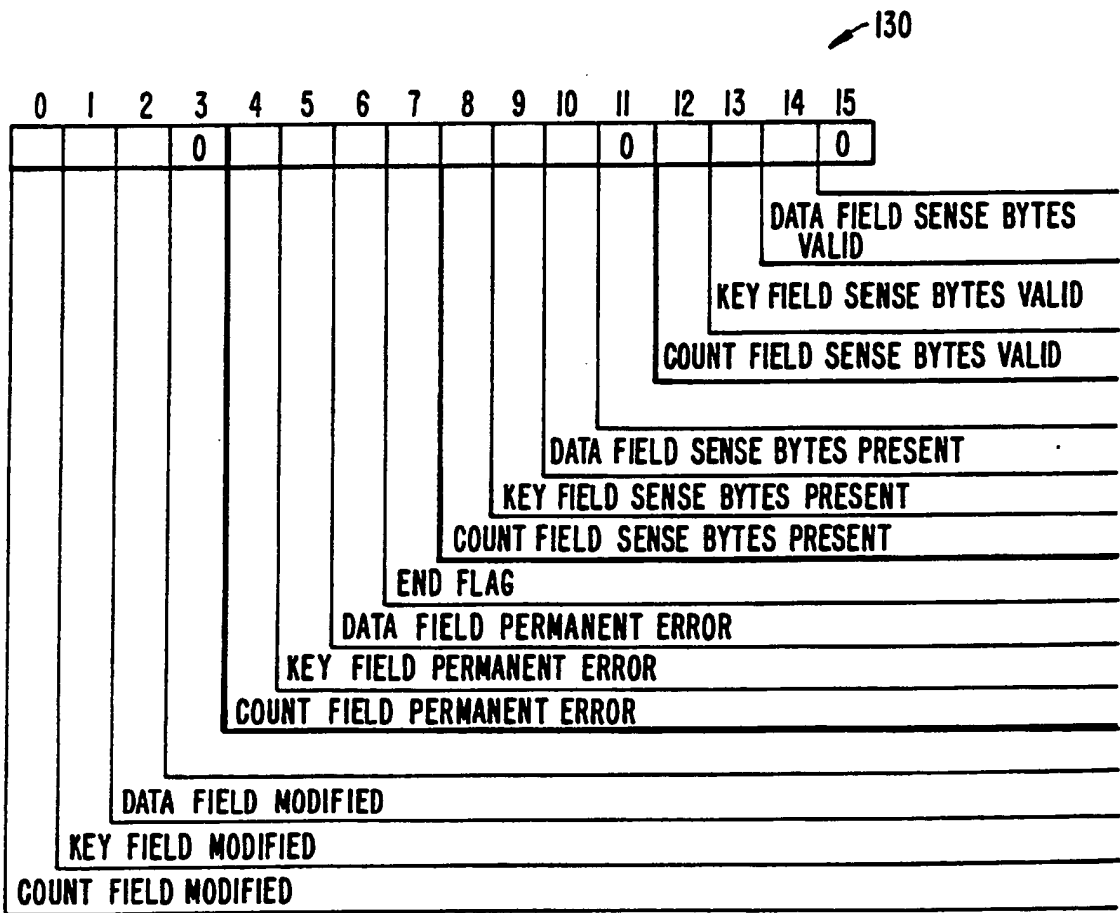


FIG. 6.

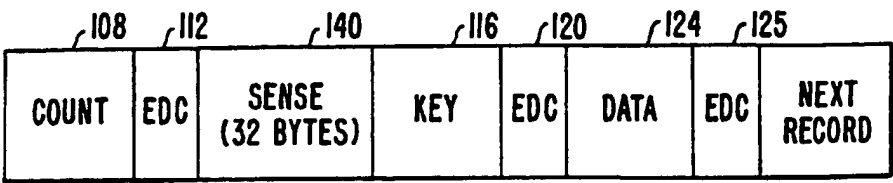
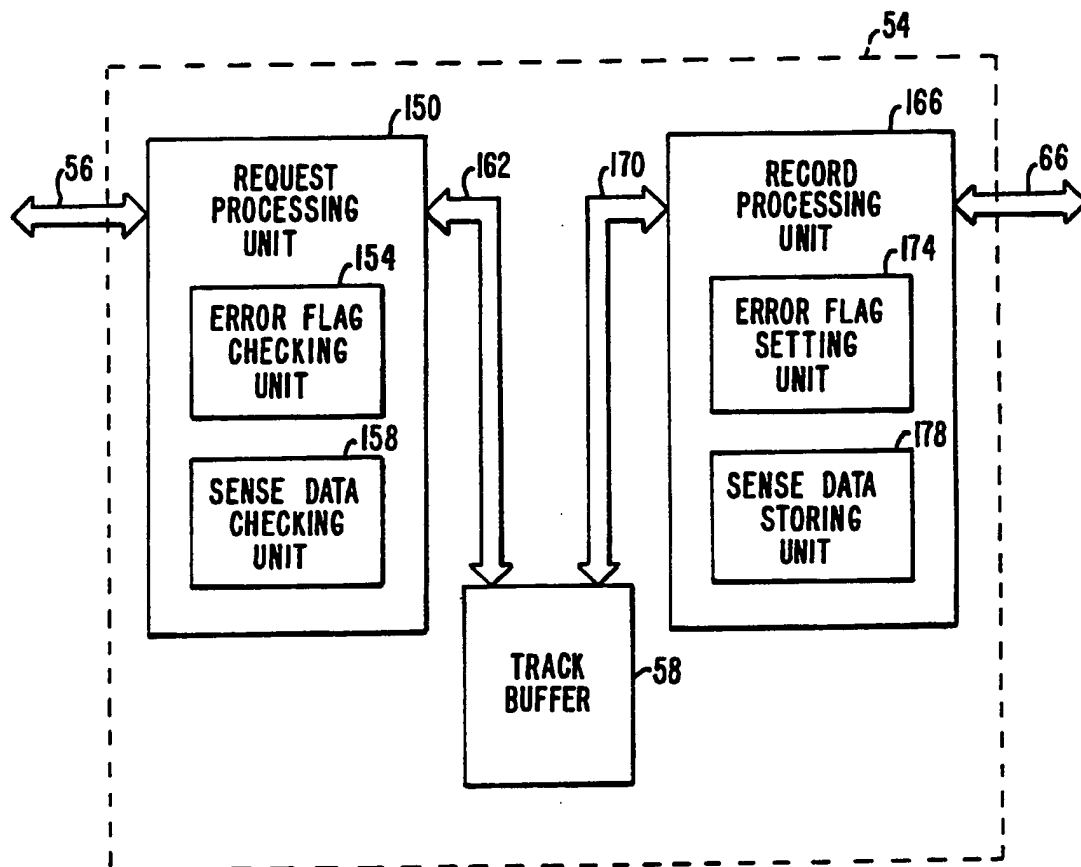


FIG. 7.

3/8

**FIG. 8.**

4/8

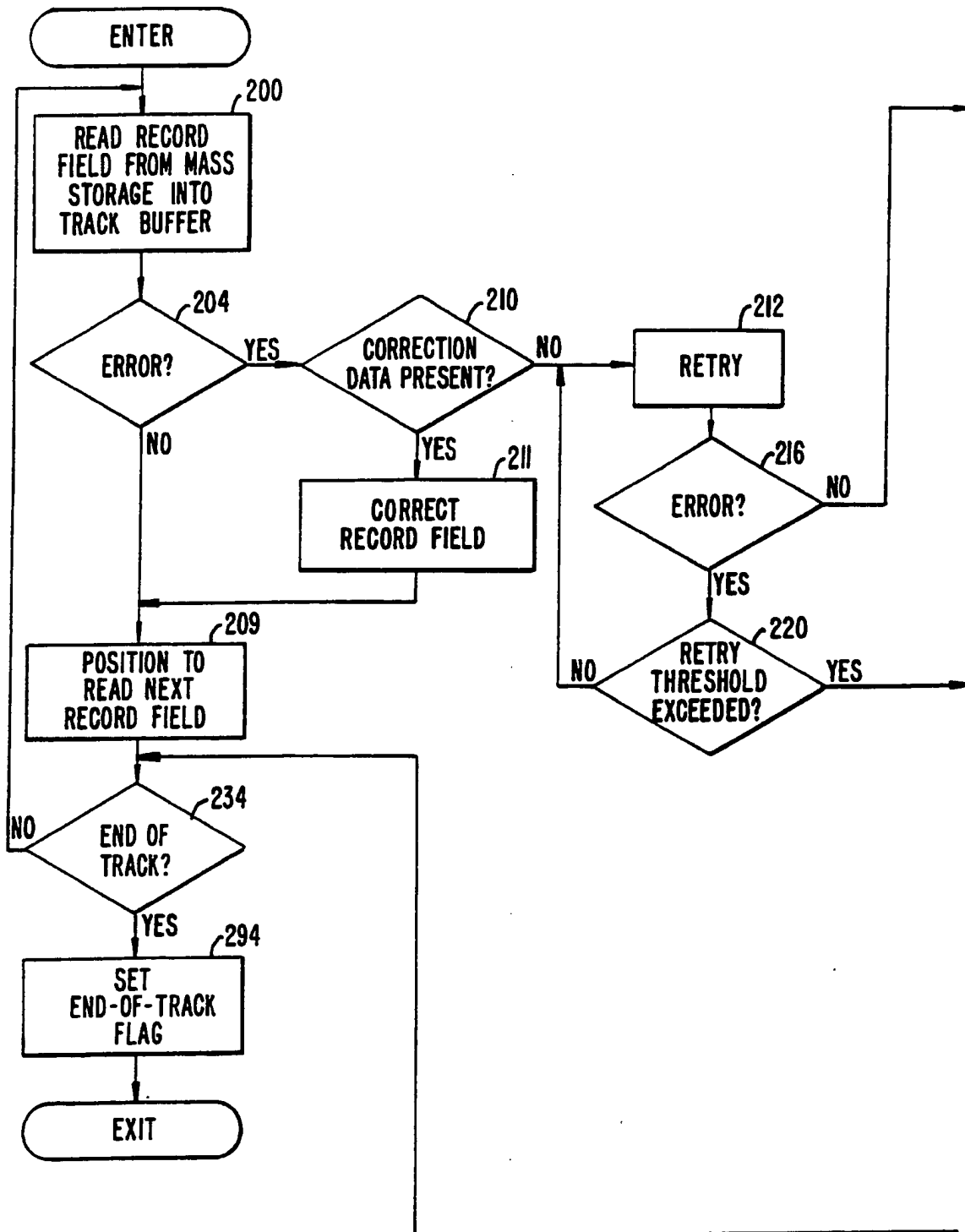


FIG. 9A.

FIG. 9A	FIG. 9B.
---------	----------

FIG. 9.

5/8

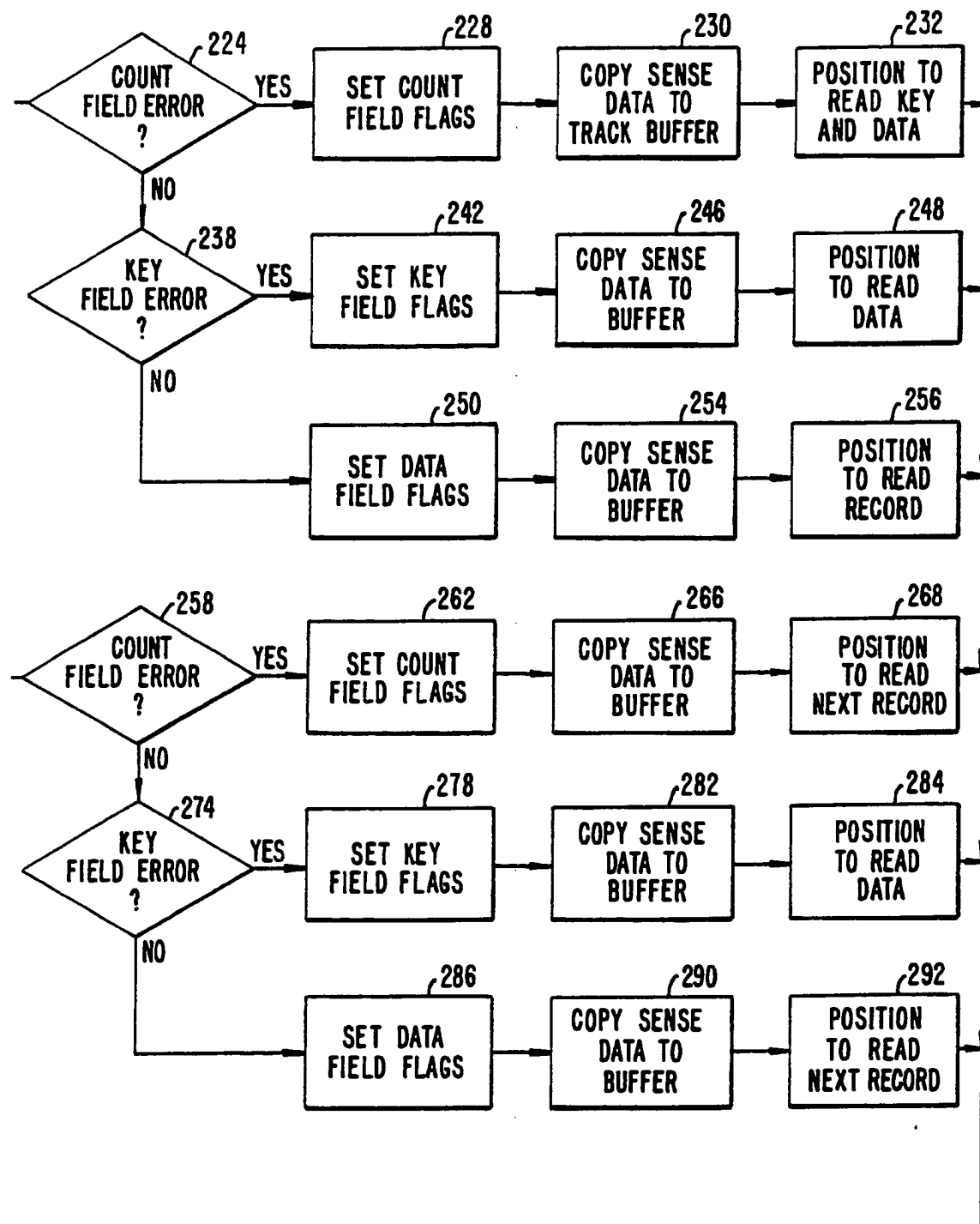


FIG. 9B.

6/8

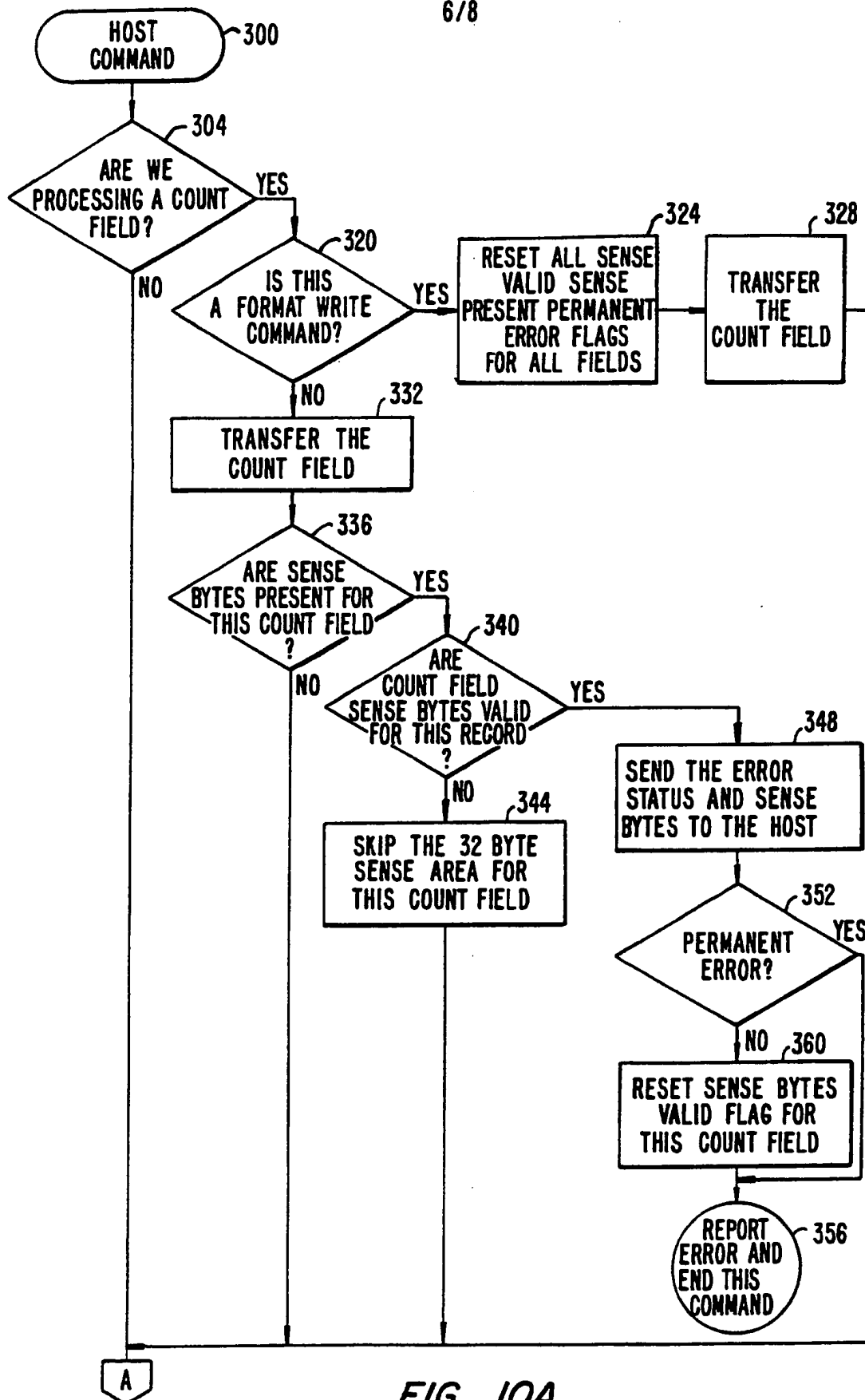
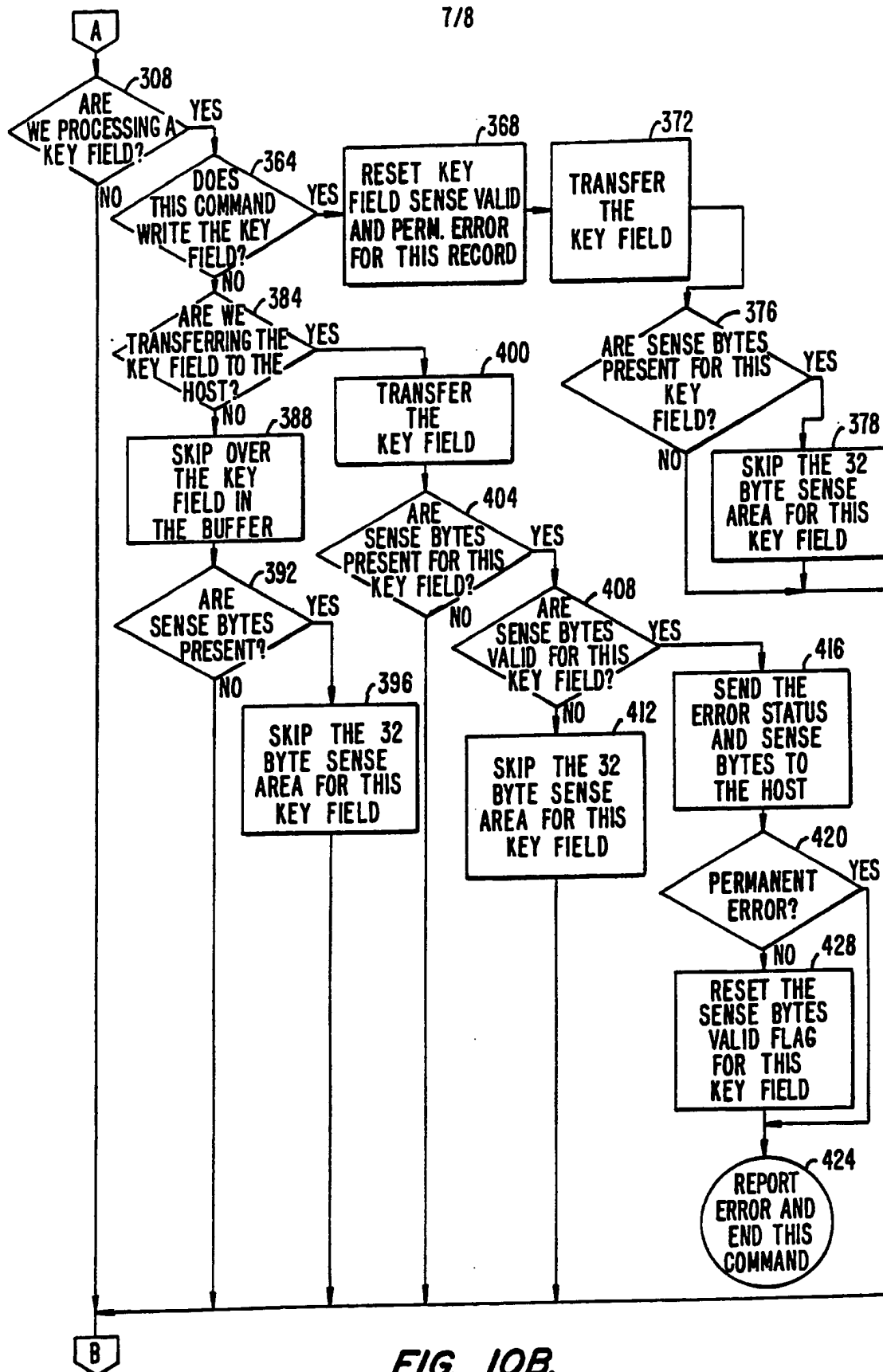


FIG. 10A.





8/8

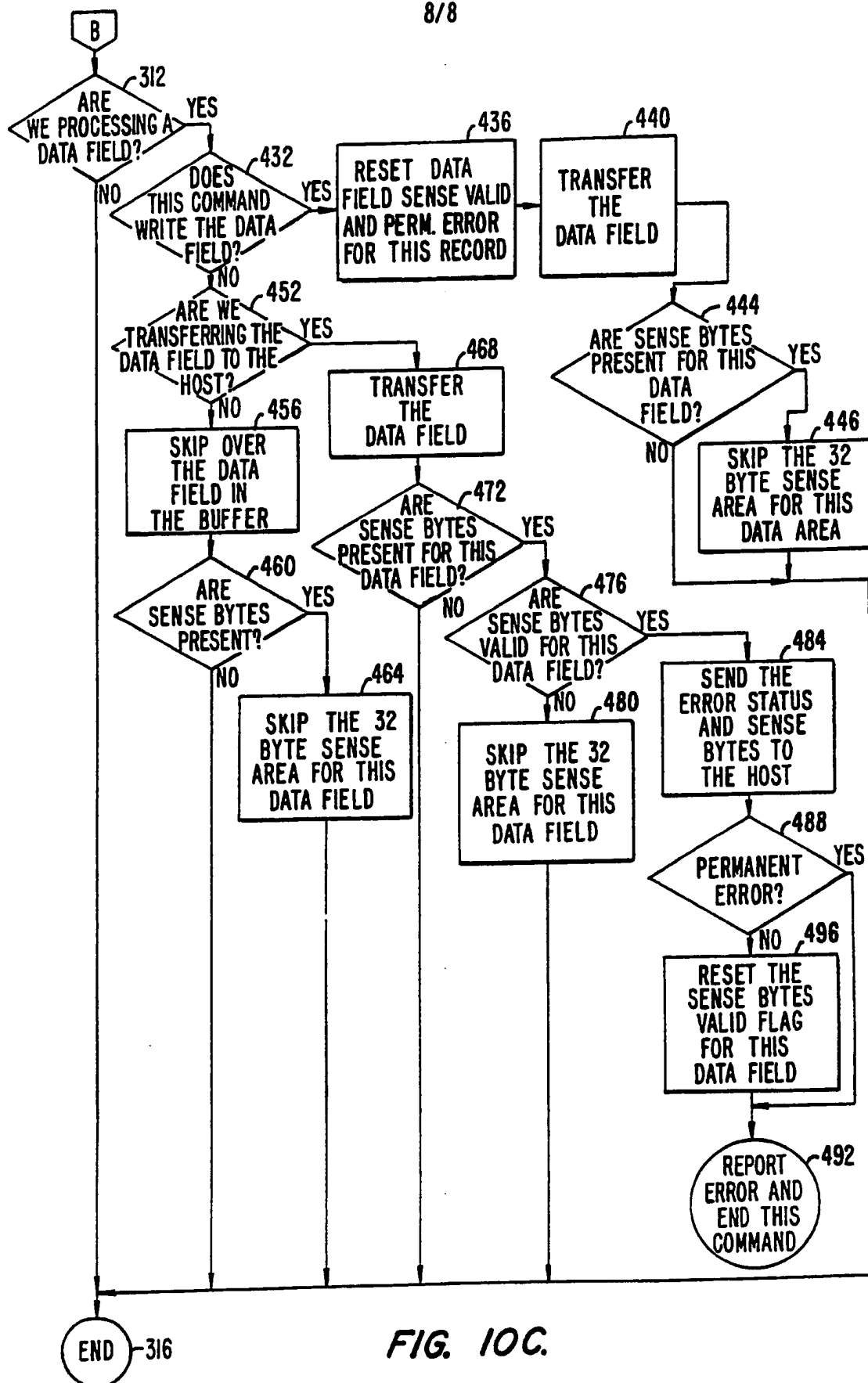


FIG. 10C.

TRACK BUFFER WITH EMBEDDED ERROR DATA

5

## BACKGROUND OF THE INVENTION

This invention relates to computer systems and, more particularly, to systems for managing data errors.

10 Data stored on a mass storage device may be corrupted due to physical damage of the storage media, by electromagnetic radiation, or by other factors. Additionally, valid data stored on the mass storage device may be corrupted by the accessing mechanism due to hardware failure, etc. Some data errors may be corrected. For example, the error may be a transient one and may be cured  
15 by rereading the data from the storage media. Other data errors may be cured using various error correction algorithms. Once the error is corrected, the data may be provided to the user as if no error occurred, or else the fact that an error occurred may be reported to the user to  
20 alert the user that more serious problems may lie ahead. Data errors which cannot be corrected are termed "permanent errors," and the user typically is notified of the permanent error without being provided with the corrupted data.

Many computer systems employ cache memories for  
25 buffering data from a mass storage device so that a host processor may obtain the stored data more rapidly than if the host processor obtained the data directly from the mass storage device. Ordinarily, only error-free data is stored in the cache memory. Thus, if an error (whether correctable  
30 or permanent) occurred when data to be stored in the cache memory was read from the mass storage device, then the host processor would be forced to access the data directly from the mass storage device, whereupon it would learn of the error. That means the host processor must spend a  
35 substantial amount of time attempting to access data that it will not get.

## SUMMARY OF THE INVENTION

The present invention is directed to a computer system having data error information stored in a cache memory so that the host processor may learn of the error without having to access a mass storage device. In one embodiment of the present invention, a buffer stores a plurality of records read from a mass storage device. The buffer includes a record information field and an error flag field for each stored record. The record information field includes count, key (i.e., record access information) and record data fields, together with an optional error data field for storing error data indicating a characteristic of an error which occurred when the record was read from the mass storage device. The error data field is typically located adjacent to the field which is in error.

The error flag field stores information indicating whether an error occurred when the record was read from the mass storage device. More particularly, the error flag field includes a count error field for storing information indicating whether an error occurred when the count information was read from the mass storage device, a key error field for storing information indicating whether an error occurred when the record access information was read from the mass storage device, and a record data error field for storing information indicating whether an error occurred when the record data was read from the mass storage device. The error flag field also indicates whether a count, key or data error was a permanent error.

When the host processor requests from the buffer a record which has an error associated with it, the host is informed that the error occurred and is provided with the error data and the record data. If the error was a permanent error, then a unit check is provided to the host for indicating that the record data is not valid.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a particular embodiment of a computer system according to the present invention;

5 Fig. 2 is a block diagram of a particular embodiment of the intermediate processor shown in Fig. 1;

Fig. 3 is a block diagram of a particular embodiment of the track buffer shown in Fig. 2;

10 Fig. 4 is a block diagram of a particular embodiment of the track data area shown in Fig. 3;

Fig. 5 is a block diagram of a particular embodiment of a track record shown in Fig. 4;

Fig. 6 is a diagram showing a particular embodiment of a record flag shown in Fig. 4;

15 Fig. 7 is a block diagram of a particular embodiment of a track record which has error data stored therein;

Fig. 8 is a block diagram of a particular embodiment of the main processor shown in Fig. 2;

20 Fig. 9 is a flow chart showing how error data is stored in the track buffer; and

Figs. 10A-C are flow charts showing how host record requests are processed.

## 25 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 is a block diagram of a particular embodiment of a computer system 10 according to the present invention. Computer system 10 includes a host processor 14, an intermediate processor 18, a mass storage control unit 26, and a mass storage device 34. Intermediate processor 18 stores data from mass storage device 34 for rapid access by host 14. Host processor 14 communicates with intermediate processor 18 through a host-processor communication path 22. Intermediate processor 18 is coupled to mass storage control unit 26 through a processor-control communication path 30, and mass storage control unit 26 is coupled to mass storage device 34 through a control-storage communication path 38.

30

35

4

Fig. 2 is a block diagram of a particular embodiment of intermediate processor 18. In this embodiment, intermediate processor 18 is constructed in accordance with the teachings of copending U.S. patent application Serial No. 08/018,635, entitled Intermediate Processor Disposed Between a Host Channel and a Storage Director, incorporated herein by reference. As shown in Fig. 2, intermediate processor 18 includes a control unit interface circuit 42 which receives data requests from host 14 through host-processor communication path 22; a front end circuit 46 which communicates with control unit interface 42 through a communication path 50 for processing the requests received by control unit interface 42; a main processor 54 which communicates with front end circuit 46 through a communication path 56 and which manages data stored in a track buffer 58; and a channel interface circuit 62 for communicating data between mass storage control unit 26 and main processor 54 through communication paths 30 and 66.

Fig. 3 is a block diagram showing the logical organization of track buffer 58. Track buffer 58 includes a page pointer area 70 for storing a plurality of pointers to pages which form the track buffer; a chain pointer area 74 for storing device chain pointers (for all tracks associated with the device), LRU pointers, sub-buffer chain pointers (which are used when there is more than one buffer describing the same track), etc; an operating system work area 78; and a track data area 82 for storing a track of data (comprising a plurality of records) from mass storage device 34.

Fig. 4 is a block diagram of a particular embodiment of track data area 82. Track data area 82 comprises a buffer modification flag 86; a Cylinder/Cylinder/Head/Head (CCHH) area 90 which identifies the track so that the task accessing the track may compare its internal CCHH to verify that it has the correct track; a record flag area 94 for storing status information about the

plurality of records in the track; and a track record area 98 for storing the actual track data and control information.

Fig. 5 is a block diagram of a particular embodiment of a track record 100 that is stored in track record area 98. Track record 100 includes a home address field 104; a 2-byte error detection code (EDC) for home address field 104; an 8-byte count field 108 for indicating the length of the record in bytes; a 2-byte EDC 112 for count field 108; an optional key field 116 (up to 255 bytes) whose length is determined by count field 108 for providing access information for the record; a 2-byte EDC 120 for key field 116; a variable-length data field 124; and a 2-byte EDC 125 for data field 124. Following EDC 125 are the count, key, and data fields (together with their EDC values) for the remaining records in the track.

Fig. 6 is a diagram of a particular embodiment of a record flag 130 from record flag area 94. There is one such record flag for each record stored in track record area 98. Bits [0:2] store information indicating whether the count, key or data field, respectively, have been modified. Bits [4:6] store information indicating whether a permanent error occurred when the count, key or data fields, respectively, were read from the mass storage device. Bit [7] functions as an END-OF-TRACK flag for indicating that the corresponding record is the last record in the track. Bits [8:10] store information indicating whether error data for the count, key and/or data fields, respectively, are present in the record. Finally, bits [12:14] indicate whether the error data for the count, key and/or data fields, respectively, are valid in the associated record and need to be presented to the host.

In this embodiment, the error data may comprise sense data bytes received as described in the IBM ESA/370 Principles of Operation manual, incorporated herein by reference, and the sense bytes are embedded within the track record. Generally, the sense bytes are stored adjacent to

the field which caused the data error. For example, Fig. 7 shows a sense data byte field 140 between count field 108 and key field 116 for indicating an error occurred when count field 108 was read from mass storage. As used in this example (and in the claims), the term "count field" may mean the bytes comprising the count value itself, or the bytes comprising both the count value and its associated EDC. The same is true for the key and data fields. If key field 116 caused the error, then the sense bytes would be disposed between key field 116 and data field 124. If a data field caused the error, then the sense bytes would be disposed directly after data field 124.

When there is no count field 108 (because of a permanent error when reading the count field), then there will be no key or data fields. When there is a count field, there may or may not be key or data fields depending on whether errors occurred when reading those fields from mass storage device 34 and whether or not the errors were permanent. In general, if the field data cannot be read after ten retry attempts, then the appropriate permanent error flag is set. If the count field was successfully read, then the count field may be used to skip over the failing record and subsequent records may be read until a good portion of the track is found. If the count field was not successfully read, then either the remainder of the track may be ignored, selected portions of the track may be skipped, or a search made for a reliable indicator of the next record (such as a predetermined data sequence). The remaining good portions of the track then may be staged into track record area 98.

Fig. 8 is a block diagram of portions of main processor 54 relevant to the present invention. A request processing unit 150 is coupled to front end circuit 46 through communication path 56 and to track buffer 58 through a communication path 162. Request processing unit 150 includes an error flag checking unit 154 for determining whether an error occurred when the requested record was read

7

from mass storage device 34, and a sense data processing unit 158 for selectively communicating any sense data stored in the requested record to front end circuit 46. A record processing unit 166 is coupled to track buffer 58 through a communication path 170 and to channel interface 62 through a communication path 66. Record processing unit 166 includes an error flag setting unit 174 for setting the appropriate error flag when an error is reported by mass storage control unit 26, and a sense data storing unit 178 for storing sense data received from mass storage control unit 26 in the appropriate location in the record.

Fig. 9 is a flow chart showing the operation of record processing unit 166. Record processing unit 166 requests mass storage control unit 26 to read a record field (count, key or data as determined by the position of the read head) from mass storage device 34 into track record area 98 of track buffer 58 in a step 200. It is then ascertained in a step 204 whether an error occurred when the requested record field was read from mass storage device 34. If not, then the mass storage device 34 is positioned to read the next record field in a step 209. If an error occurred, it is then ascertained whether correction data is present (for those systems which provide such) in a step 210. If so, then the record field is corrected in a step 211, and processing continues in step 209. If no correction data is present, then record processing unit 166 asks mass storage control unit 26 to retry the read in a step 212. It is then ascertained in a step 216 whether another error occurred. If so, it is then ascertained in a step 220 whether the retry threshold (e.g., 10) has been exceeded. If not, then the retries continue in step 212.

If it is ascertained in step 216 that the retry was successful, then it is ascertained in a step 224 whether the count field caused the original error. If so, then the count field sense bytes present and valid flags (Fig. 6) are set in a step 228, the sense data received from mass storage control unit 26 is stored adjacent to the count field in a



8

step 230, and mass storage device 34 is positioned to read the key and data fields for the record in a step 232. It is then ascertained in a step 234 whether the end of the track has been reached. This typically will not be the case when the count and key fields are being processed, so processing continues in step 200.

If it is ascertained in step 224 that the error did not occur from reading the count field, it is then ascertained in a step 238 whether the error occurred as a result of reading the key field. If so, then the key field sense bytes present and valid flags are set in a step 242, the sense data is stored adjacent to the key field in a step 246, mass storage device 34 is positioned to read the data field for the record in a step 248, and processing continues in step 234. If the key field did not cause the error, then the data field sense bytes present and valid flags are set in a step 250, the sense data is stored adjacent to the data field in a step 254, mass storage device is positioned to read the count field of the next record in a step 256, and processing continues in step 234.

If it is ascertained in step 220 that the retry threshold has been exceeded, then a permanent error condition exists. Therefore, it is ascertained in a step 258 whether the error occurred as a result of reading the count field. If so, then the count field permanent error and sense bytes valid and present flags are set in a step 262, the sense data is stored adjacent to the count field in a step 266, mass storage device 34 is positioned to read the next record in a step 268, and processing continues in step 234. If the count field did not generate the error, it is then ascertained in a step 274 whether the key field caused the error. If so, the key field permanent error and sense bytes present and valid flags are set in a step 278, the sense data is stored adjacent to the key field in a step 282, mass storage device 34 is positioned to read the data for the record in a step 284, and processing continues in step 234. If the key field did not cause the error, then

the data field permanent error and sense bytes present and valid flags are set in a step 286, the sense data is stored adjacent to the data field in a step 290, mass storage device 34 is positioned to read the next record in a step 292, and processing continues in step 234.

When it is ascertained in step 234 that the end of the track has been reached, then the END-OF-TRACK flag is set in a step 294, and the program exits.

Figures 10A-C are flow charts showing the operation of request processing unit 150. Request processing unit 150 receives a host command record request in a step 300 and ascertains in a step 304 whether the count field is being processed. If not, it is then ascertained in a step 308 (Figure 10B) whether a key field is being processed. If not, it is then ascertained in a step 312 (Figure 10C) whether a data field is being processed. If not, then the command is terminated in a step 316. Typically one or more of steps 304, 308 or 312 provide an affirmative response. Thus, if it is ascertained in step 304 that the count field is being processed, it is then ascertained in a step 320 whether the command is a format write command. If so, then the sense bytes valid, sense bytes present and permanent error flags for all fields are reset in a step 324. The count field is then transferred in a step 328, and processing continues in step 308.

If the command was not a format write command, then the count field is transferred in a step 332, and it is then ascertained in a step 336 whether the count field sense bytes present flag is set. If not, then processing continues in step 308. If the count field sense bytes present flag is set, it is then ascertained in a step 340 whether the count field sense bytes valid flag is set. If not, then the 32 bytes are skipped in a step 344, and processing continues to step 308. If the count field sense bytes valid flag is set, then the error status and sense bytes are sent to the host in a step 348. It is then ascertained in a step 352 whether the count field permanent

10

error flag is set. If so, then the error is reported in a step 356, and the command is terminated. If the count field permanent error flag is not set, then the count field sense bytes valid flag is reset in a step 360, the error is reported in step 356, and the command terminates.

If it is ascertained in step 308 that a key field is being processed, it is then ascertained in a step 364 whether the command is one to write the key field. If so, then the key field sense bytes valid and key field permanent error flags are reset in a step 368, and the key field is transferred in a step 372. It is then ascertained in a step 376 whether the key field sense bytes present flag is set. If not, then processing continues in step 312. If the key field sense bytes present flag is set, then the 32 sense bytes are skipped in a step 380, and processing continues in step 312.

If it is ascertained in step 364 that the command does not write the key field, it is then ascertained in a step 384 whether the key field is being transferred to the host. If not, then the key field is skipped in a step 388. It is then ascertained in a step 392 whether the key field sense bytes present flag is set. If not, then processing continues in step 312. If the key field sense bytes present flag is set, then the 32 sense bytes are skipped in a step 396, and processing continues in step 312.

If it is ascertained in step 384 that the key field is being transferred to the host, then the key field is transferred in a step 400. It is then ascertained in a step 404 whether the key field sense bytes present flag is set. If not, then processing continues in step 312. If the key field sense bytes present flag is set, it is then ascertained in a step 408 whether the key field sense bytes valid flag is set. If not, then the 32 sense bytes are skipped in a step 412, and processing continues in step 312. If the key field sense bytes valid flag is set, then the error status and sense bytes are sent to the host in a step 416. It is then ascertained in a step 420 whether the key

11

field permanent error flag is set. If so, then the error is reported in a step 424, and the command terminates. If the key field permanent error flag is not set, then the key field sense bytes valid flag is reset in a step 428, the error is reported in step 424, and the command terminates.

If it is ascertained in step 312 that the data field is being processed, then it is ascertained in a step 432 whether the command is writing the data field. If so, then the data field sense bytes valid and data field permanent error flags are reset in a step 436, and the data field is transferred in a step 440. It is then ascertained in a step 444 whether the data field sense bytes present flag is set. If not, then the command is terminated in step 316. If the data field sense bytes present flag is set, then the 32 sense bytes are skipped in a step 448, and processing terminates in step 316.

If it is ascertained in step 432 that the command does not write the data field, it is then ascertained in a step 452 whether the data field is being transferred to the host. If not, then the data field is skipped in a step 456. It is then ascertained in a step 460 whether the data field sense bytes present flag is set. If not, then the command terminates in step 316. If the data field sense bytes present flag is set, then the 32 sense bytes are skipped in step 464, and the command terminates in step 316.

If it is ascertained in step 452 that the data field is being transferred to the host, then the data field is transferred in a step 468. It is then ascertained in a step 472 whether the data field sense bytes present flag is set. If not, then the command terminates in step 316. If the data field sense bytes valid flag is set, it is then ascertained in a step 476 whether the data field sense bytes valid flag is set. If not, then the 32 sense bytes are skipped in a step 480, and the command terminates in step 316. If the data field sense bytes valid flag is set, then the error status and sense bytes are sent to the host in a step 484. It is then ascertained in a step 488 whether the

12

data field permanent error flag is set. If so, then the error is reported to the host in a step 492, and th command terminates. If the data field permanent error flag is not set, then the data field sense bytes valid flag is reset in  
5 a step 496, the error is reported to the host in step 492, and the command terminates.

While the above is a complete description of a preferred embodiment of the present invention, various modifications may be employed. Consequently, the scope of  
10 the invention should not be limited except as described in the claims.

## CLAIMS:

1. A buffer for storing a plurality of records read from a mass storage device, the buffer comprising, for each record read from the mass storage device:
- an error flag field for storing information indicating whether an error occurred when the record was read from the mass storage device; and
- a record information field for storing record information for the record read from the mass storage device, the record information field including an error data field for storing error data indicating a characteristic of an error which occurred when the record was read from the mass storage device.
2. The buffer according to claim 1 wherein the error data field exists in a record only when an error occurred when the record was read from the mass storage device.
3. The buffer according to claim 1 wherein each record information field further comprises:
- a key field for storing access information for the record read from the mass storage device;
- a record data field for storing record data read from the mass storage device; and
- a count field for storing count information read from the mass storage device, the count information indicating a length of the record data field read from the mass storage device.
4. The buffer according to claim 3 wherein each error flag field comprises:
- a count error field for storing information indicating whether the error occurred when the count information was read from the mass storage device;

14

a key error field for storing information indicating whether the error occurred when the access information was read from the mass storage device; and

5 a record data error field for storing information indicating whether the error occurred when the record data was read from the mass storage device.

10 5. The buffer according to claim 4 wherein the error data field is disposed adjacent to the count field when the error occurred when the count information was read from the mass storage device.

15 6. The buffer according to claim 4 wherein the error data field is disposed adjacent to the key field when the error occurred when the access information was read from the mass storage device.

20 7. The buffer according to claim 4 wherein the error data field is disposed adjacent to the record data field when the error occurred when the record data was read from the mass storage device.

25 8. The buffer according to claim 4 wherein the key field is disposed between the count field and the record data field, wherein the error data field is disposed between the count field and the key field when the error occurred when the count information was read from the mass storage device, and wherein the error data field is disposed between the key field and the data field when the error occurred  
30 when the access information was read from the mass storage device.

35 9. An apparatus for accessing data from a mass storage device comprising:  
record reading means for reading a record from the mass storage device;

15

error detecting means, coupled to the record reading means, for detecting when an error occurred when the record was read from the mass storage device;

5 a buffer for storing a plurality of records read from the mass storage device, the buffer comprising, for each record read from the mass storage device:

an error flag field for storing information indicating whether an error occurred when the record was read from the mass storage device;

10 a record information field for storing record information for the record read from the mass storage device, the record information field including an error data field for storing error data indicating a characteristic of an error which occurred when the record was read from the mass storage device;

15 record information storing means, coupled to the record reading means and to the buffer, for storing record information read from the mass storage device in the record information field;

20 error flag setting means, coupled to the buffer and to the error detecting means, for storing information in the error flag field indicating an error occurred when the corresponding record was read from the mass storage device; and

25 error data storing means, coupled to the buffer and to the error detecting means, for storing error data in the error data field when an error occurred when the corresponding record was read from the mass storage device.

30 10. The apparatus according to claim 9 wherein the error data field exists in the record information field only when an error occurred when the corresponding record was read from the mass storage device.

35 11. The apparatus according to claim 9 wherein each record information field further comprises a record data



16

field for storing record data read from the mass storage device.

12. The apparatus according to claim 11 further  
5 comprising:  
request receiving means for receiving record requests  
from a data processing element; and  
response means, coupled to the buffer and to the  
request receiving means, for communicating the record data  
10 corresponding to a requested record to the data processing  
element when the error flag indicates no error occurred when  
the requested record was read from the mass storage device.

13. The apparatus according to claim 12 wherein the  
15 response means includes means for communicating the error  
data corresponding to the requested record to the data  
processing element when the error flag field indicates an  
error occurred when the requested record was read from the  
mass storage device.

20  
14. The apparatus according to claim 13 wherein the  
response means includes means for communicating the record  
data corresponding to the requested record to the data  
processing element when the error flag field indicates an  
25 error occurred when the requested record was read from the  
mass storage device.

15. The apparatus according to claim 12 wherein the  
buffer further comprises, for each record read from the mass  
30 storage device, an error data status field for storing  
information indicating whether the error data for the  
corresponding record has been communicated to the data  
processing element.

35 16. The apparatus according to claim 15 further  
comprising error data status storing means, coupled to the  
error data status field and to the response means, for

17

storing information in the error data status field  
indicating the error data for the corresponding record has  
been communicated to the data processing element when the  
response means communicates the error data for the  
5 corresponding record to the data processing element.

17. The apparatus according to claim 11 wherein the  
record information field further comprises:

10 a count field for storing count information read from  
the mass storage device, the count information indicating a  
length of the record data field read from the mass storage  
device; and

a key field for storing access information for the  
record read from the mass storage device.

15

18. The apparatus according to claim 17 wherein each  
error flag field comprises:

20 a count error field for storing information indicating  
whether the error occurred when the count information was  
read from the mass storage device;

a key error field for storing information indicating  
whether the error occurred when the access information was  
read from the mass storage device; and

25 a record data error field for storing information  
indicating whether the error occurred when the record data  
was read from the mass storage device.

19. The buffer according to claim 18 wherein the error  
data field is disposed adjacent to the count field when the  
30 count error field indicates the error occurred when the  
count information was read from the mass storage device.

20. The buffer according to claim 18 wherein the error  
data field is disposed adjacent to the key field when the  
35 key error field indicates the error occurred when the access  
information was read from the mass storage device.

21. The buffer according to claim 18 wherein the error data field is disposed adjacent to the record data field when the record data error field indicates the error occurred when the record data was read from the mass storage device.

22. The buffer according to claim 18 wherein the key field is disposed between the count field and the record data field, wherein the error data field is disposed between the count field and the key field when the count error field indicates the error occurred when the count information was read from the mass storage device, and wherein the error data field is disposed between the key field and the data field when the key error field indicates the error occurred when the access information was read from the mass storage device.

23. A method for accessing data from a mass storage device comprising the steps of:

- 20 reading a record from the mass storage device;
- storing information in an error flag field corresponding to the read record in a buffer, the error flag field indicating whether an error occurred when the record was read from the mass storage device;
- 25 storing error data in an error data field corresponding to the read record in the buffer when an error occurred when the record was read from the mass storage device, the error data field indicating a characteristic of the error which occurred when the record was read from the mass storage device;
- 30 storing record data in a record field corresponding to the read record in the buffer;
- receiving a record request from a data processing element;
- 35 determining whether the error flag field corresponding to the requested record indicates an error occurred when the requested record was read from the mass storage device; and

communicating the stored record data corresponding to the requested record to the data processing element when the error flag field indicates no error occurred when the requested record was read from the mass storage device.

5

24. The method according to claim 23 further comprising the step of communicating the error data corresponding to the requested record to the data processing element when the error flag field indicates an error occurred when the requested record was read from the mass storage device.

10

25. The method according to claim 24 further comprising the step of communicating the record data corresponding to the requested record to the data processing element when the error flag field indicates an error occurred when the requested record was read from the mass storage device.

15

26. The method according to claim 23 further comprising the step of storing information in a error data status field in the buffer, the information indicating whether the error data for a corresponding record has been communicated to the data processing element.

20

25

27. The method according to claim 32 further comprising the step of storing information in the error flag field indicating whether an error which occurred when the record was read from the mass storage device was a permanent error.

30

20

28. Apparatus for accessing data from a mass storage device, the apparatus being substantially as herein described with reference to the accompanying drawings.

5           29. A method of accessing data from a mass storage device, the method being substantially as herein described with reference to the accompanying drawings.

<b>Patents Act 1977</b> <b>Examiner's report to the Comptroller under Section 17</b> <b>(The Search report)</b>	Application number GB 9403001.2
<b>Relevant Technical Fields</b>	Search Examiner PAUL NICHOLLS
(i) UK Cl (Ed.M)      G4A AEX, AME (ii) Int Cl (Ed.5)      GO6F 11/00, 11/08, 11/10, 11/14	Date of completion of Search 3 MAY 1994
<b>Databases (see below)</b> (i) UK Patent Office collections of GB, EP, WO and US patent specifications.  (ii) ONLINE DATABASES: WPI	Documents considered relevant following a search in respect of Claims :- 1-29

**Categories of documents**

- |   |  |
|---|--|
| <b>X:</b> Document indicating lack of novelty or of inventive step.<br><br><b>Y:</b> Document indicating lack of inventive step if combined with one or more other documents of the same category.<br><br><b>A:</b> Document indicating technological background and/or state of the art. | <b>P:</b> Document published on or after the declared priority date but before the filing date of the present application.<br><br><b>E:</b> Patent document published on or after, but with priority date earlier than, the filing date of the present application.<br><br><b>&amp;:</b> Member of the same patent family; corresponding document. |
|---|--|

Category	Identity of document and relevant passages	Relevant to claim(s)
X	GB 2202976 A (IBM) see page 2 line 19 - page 3 line 4	1, 9, 23-27

**Databases:** The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).